

10830-086001

日 本 国 特 許 庁  
JAPAN PATENT OFFICE



別紙添付の書類に記載されている事項は下記の出願書類に記載されている事項と同一であることを証明する。

This is to certify that the annexed is a true copy of the following application as filed with this Office

出 願 年 月 日

Date of Application:

2001年 1月10日

出 願 番 号

Application Number:

特願2001-002029

出 願 人

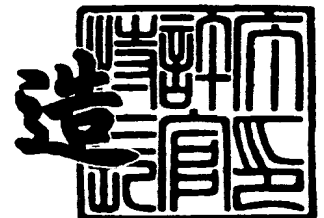
Applicant(s):

安藤電気株式会社

2001年12月14日

特 許 庁 長 官  
Commissioner,  
Japan Patent Office

及 川 耕 造



出証番号 出証特2001-3108615

【書類名】 特許願

【整理番号】 S00-12-17

【あて先】 特許庁長官 殿

【国際特許分類】 G05F 15/00

【発明者】

【住所又は居所】 東京都大田区蒲田4丁目19番7号 安藤電気株式会社  
内

【氏名】 田内 正治

【特許出願人】

【識別番号】 000117744

【氏名又は名称】 安藤電気株式会社

【代理人】

【識別番号】 100099195

【弁理士】

【氏名又は名称】 宮越 典明

【選任した代理人】

【識別番号】 100116182

【弁理士】

【氏名又は名称】 内藤 照雄

【手数料の表示】

【予納台帳番号】 030889

【納付金額】 21,000円

【提出物件の目録】

【物件名】 明細書 1

【物件名】 図面 1

【物件名】 要約書 1

【包括委任状番号】 9909752

【包括委任状番号】 0014291

【プルーフの要否】 要

【書類名】 明細書

【発明の名称】 高速 F F T 演算処理方法

【特許請求の範囲】

【請求項 1】 多数の被 F F T データを F F T 演算する高速 F F T 演算処理方法において、

前記多数の被 F F T データ (N) を、F F T 演算処理に用いるメモリのアクセスに適したブロックに分割するステップ(a)と、

前記メモリに対して、前記ブロックに分割されたデータを順次転送するステップ(b)と、

前記メモリに転送された F F T データに基づく F F T 演算処理を実行するステップ(c)と、

前記ステップ(c)及びステップ(c)を、全ての分割されたブロックについて繰り返すステップ(d)と、を有することを特徴とする高速 F F T 演算処理方法。

【請求項 2】 前記多数の被 F F T データの F F T 演算処理を複数のステージに分割し、各ステージ間で、被 F F T データの並び替えを実行するステップを有することを特徴とする請求項 1 に記載の高速 F F T 演算処理方法。

【請求項 3】 前記ステップ(a)における被 F F T データの分割は、前記メモリアクセスに際して、メモリを構成するバンクの再設定が不要なデータ範囲内になるように行われることを特徴とする請求項 1 又は 2 に記載の高速 F F T 演算処理方法。

【請求項 4】 前記ステップ(a)における被 F F T データの分割は、前記メモリアクセスに際して、ローアドレス又はカラムアドレスの再設定が不要なデータ範囲内になるように行われることを特徴とする請求項 1 又は 2 に記載の高速 F F T 演算処理方法。

【請求項 5】 前記被 F F T データは、それぞれ実数部及び虚数部からなり、回転子との間で F F T 演算処理されることを特徴とする請求項 1 ～ 4 のいずれか 1 項に記載の高速 F F T 演算処理方法。

【請求項 6】 前記回転子が前記ブロックに対応して予めテーブルに保管されていることを特徴とする請求項 1 ～ 5 のいずれか 1 項に記載の高速 F F T 演算

処理方法。

【請求項7】 前記被FFTデータの内の虚数部データの演算を省略することを特徴とする請求項1～6のいずれか1項に記載の高速FFT演算処理方法。

【請求項8】

前記FFT演算において、回転子の実数部又は虚数部のいずれか一方がゼロの場合には、乗算を省略することを特徴とする請求項1～7のいずれか1項に記載の高速FFT演算処理方法。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】

本発明は、FFT（高速フーリエ変換）演算に適用する高速FFT演算処理方法に関する。

【0002】

【従来の技術】

信号処理においてFFT（高速フーリエ変換）、IFFT（逆フーリエ変換処理）は、例えばDSP（デジタル信号プロセッサ）等の演算デバイスの性能向上により高速に処理できるため、大容量データの解析が実用レベルで可能になった。FFT（以降、フーリエ変換も逆フーリエ変換もFFTとして扱う）では、周波数間引き、時間間引きによらず全てのデータを使用して演算するので、実部、虚部のデータ分の記憶デバイス（以降メモリ）が必要である。

【0003】

FFT演算処理はメモリに格納されている実部、虚部データを演算デバイスのレジスタへ転送し、そこで回転子との演算（乗算・加算）を行い、演算結果をメモリへ格納する。

演算式、式1は、図1中に式1FFT算出式として示されている。（FFT演算は既知の手法であるので、説明は省略する。）

この演算を全てのデータに対して繰り返し行う。

【0004】

実際に、演算デバイスで処理する場合、演算の度に、データを演算デバイス内レジスタへ転送してから演算している。

よって、転送時間の短縮、すなわちメモリへのアクセス時間の短縮はFFT演算処理の高速化につながる。

#### 【0005】

例えば、16個のデータを使用した周波数間引き方式のFFT演算で説明する。周波数間引きの場合、上記、式1の周波数間引き法を使用する。

それぞれの扱うデータは、図2-1従来処理のように、入力データの並び替え後、第一段で入力データX0と入力データX8に対して式1の計算をする。

これを全ての入力データに対して図2-1のように演算すると第1段の演算が終了する。

#### 【0006】

次に第2段として第一段の入力データX0とX8の結果と、X4とX12の演算結果に対して同様の演算をする。

この演算を同様の方法で処理し、第2段の演算を処理していき、これを第3段、第4段と進めていくと、最終的にY0からY15までの出力結果を得ることができる。

#### 【0007】

図2-2に太線で示すように、Y0のデータを求めるためにはX0～X15まで全てのデータを使用することになる。

したがって、入力データX0～X15までの全てのデータをメモリに記憶しておき、演算毎にメモリから演算デバイスへ必要なデータを転送して演算が行われていた。

#### 【0008】

パソコンなどでFFT演算する場合、データは、例えばSDRAM上にデータを配置される場合が多い。

このSDRAMは、同じバンク、同じローアドレス以内のデータを扱う場合には、カラムアドレスの変更だけでよい。そのため、データ毎のバンクまたは、ローアドレスの再設定が不要であるが、バンク、またはローアドレスの範囲を超えるデー

タを扱う場合には、バンク又はローアドレスの再設定が必要になる。

同一ローアドレスで指定できない容量のFFT演算の場合（図3参照）、例えば、第3段のX0とX1が異なるローアドレスに割り当てられたとする。

【0009】

この場合の第1のケースとして、

図5-1の様に、ローアドレスの設定のためにACTコマンドを送信し、Readコマンドによってデータを読み込むメモリアクセス方式（アクセス方式A）であれば、X0とX1のデータを読み込む場合に、それぞれ5clkを要する。（SDRAMなどに相当する方式）

【0010】

また、第2のケースとして、

図5-3の様に、ローアドレスを設定し、カラムアドレスを設定するようなメモリアクセス方式（アクセス方式B）であれば、X0とX1のデータを読み込むために、それぞれ5クロック（clk）を要する。（EDO DRAMなどに相当する方式）

【0011】

また、第3のケースとして、

CPU、DSP等の演算デバイスとして、アドレス指定方式にページ方式（絶対アドレスを指定する場合、ローアドレス命令後、カラムアドレスを指定して読み込む命令体系）を採用しているデバイスの場合には、X0とX1のデータを読み込む場合、最低でも2命令×2データ=4命令の実行が必要である。

【0012】

上記の処理はデータを読み込む毎、書き込む毎に発生する。

さらに、バンクによってコマンドを設定しなければならないメモリアクセス方式の場合には、さらにバンク設定コマンドを送信しなければならない。

例えば、X0とX1の演算において、X0とX1と回転子が、それぞれ違うバンクに配置されている場合には、

先ず、X0のバンクアドレスの設定+ローアドレスの設定後、カラムアドレス設定によりデータ（X0）を読み出す。

次に、X1のバンクアドレスの設定+ローアドレスの設定後、カラムアドレス設定によりデータ(X1)を読み出す。

さらに、回転子のバンクアドレスの設定+ローアドレスの設定後、カラムアドレス設定によりデータ(回転子)を読み出した後、(前記式1)の積和演算を実行する。

その後、X0のバンクアドレスの設定+ローアドレスの設定後、カラムアドレス設定によりデータ(X0)を書き込み、さらに、X1のバンクアドレスの設定+ローアドレスの設定後、カラムアドレス設定によりデータ(X1)を書き込む必要がある。

#### 【0013】

上記の如く、メモリへのデータの読み込み及び書込みに要する時間(クロック数)の大小は、演算処理に大きく影響する。

図1及び図2に示すような、16個のFFT演算では、この影響はそれ程ではない。しかし、実用レベルの、例えば16384個のFFT演算の場合、浮動小数点演算で1データが4バイトであるため、65536バイトのデータ格納容量が実部、虚部に必要であり、上記のデータの読み出し、書込みに要する時間の影響が大きい。

また、例えば回転子をテーブルで持つと32768バイトの格納エリアが必要であるので、複数のローアドレス領域にまたがってデータを読み込んで演算することになる。

この場合、例えば、X0とX8192番目・・・X1とX8193の演算をするなどローアドレスの再設定が頻繁に発生することになり、アドレスの再設定の時間は無視できない値になってくる。

#### 【0014】

##### 【発明が解決しようとする課題】

本発明の目的は、FFT演算にともなって生じる、メモリから演算レジスタへの演算毎のデータの転送時間を短縮することにより、FFT演算全体の処理時間を短縮することである。

本発明は、大容量のデータのFFT演算において、FFTデータを分割して演

算することによって、処理時間の短縮を実現できる。

【0015】

【課題を解決するための手段】

上記課題を解決するために、多数の被FFTデータをFFT演算する高速FFT演算処理方法において、前記多数の被FFTデータ(N)を、FFT演算処理に用いるメモリのアクセスに適したブロックに分割するステップ(a)と、前記メモリに対して、前記ブロックに分割されたデータを順次転送するステップ(b)と、前記メモリに転送されたFFTデータによるFFT演算処理を実行するステップ(c)と、前記ステップ(c)及びステップ(c)を、全ての分割されたブロックについて繰り返すステップ(d)とで、高速FFT演算処理を実行することにより、FFT演算にともなって生じる、メモリから演算レジスタへの演算毎のデータの転送時間を短縮することにより、FFT演算全体の処理時間を短縮することができる。(請求項1)

【0016】

また、前記多数の被FFTデータのFFT演算処理を複数のステージに分割し、各ステージ間で、被FFTデータの並び替えを実行するステップを有することにより、より少ないメモリでFFT演算が可能になる。(請求項2)

また、前記ステップ(a)における被FFTデータの分割は、前記メモリアクセスに際して、メモリを構成するバンクの再設定が不要なデータ範囲内になるように行うことにより、バンク構成のメモリの場合に、バンクの再設定なしにデータのアクセスが可能になるので、メモリから演算レジスタへの演算毎のデータの転送時間を短縮することにより、FFT演算全体の処理時間を短縮することができる。(請求項3)

また、前記ステップ(a)における被FFTデータの分割は、前記メモリアクセスに際して、ローアドレス又はカラムアドレスの再設定が不要なデータ範囲内になるように行うことにより、一度設定した、ローアドレス又はカラムアドレスの再設定なしに、データのアクセスが可能になるので、メモリから演算レジスタへの演算毎のデータの転送時間を短縮することにより、FFT演算全体の処理時間を短縮することができる。(請求項4)



【0017】

また、前記被FFTデータは、それぞれ実数部及び虚数部からなり、回転子との間でFFT演算処理される。（請求項5）

また、前記回転子が前記ブロックに対応して予めテーブルに保管されていることにより、より高速なFFT演算処理が可能になる。（請求項6）

【0018】

また、前記被FFTデータの内の虚数部データの演算を省略する。（請求項7）

また、前記FFT演算において、回転子の実数部又は虚数部のいずれか一方がゼロの場合には、乗算を省略することにより、更に高速のFFT演算処理が可能になる。（請求項8）

この場合、現実的な信号解析において、サンプリングしたデータの周波数解析を行う場合、虚数データの演算を省略するとか、回転子の値が実部=1，虚部=0、または、実部=0、虚部=1であることにより乗算を省略する事ができるとか、等の既知の高速化手法を反映させやすい事を特徴とする。

【0019】

【発明の実施の形態】

以下、従来と同じハードウェア構成の場合でも、高速にFFT演算処理をすることが可能な、本発明のFFT高速アルゴリズムの原理を説明する。

従来技術の技術では、図3、図5-1、図5-3を用いて説明したように、メモリから演算デバイスのレジスタへのデータの転送時間が（例えば）5クロック（clk）必要である。

しかし、同1ブロックで処理できるデータを同一のローアドレスになるようなワークエリアに転送した後、当該ブロック内でのFFT演算を実行する。

このようにすると、データ転送においては、メモリアクセス方式Aの場合、最初は4クロックは必要であるが、最初の設定さえしてしまえば、その後は、1クロック（clk）でデータのreadが可能である。（図5-2参照）

また、メモリアクセス方式Bにおいても同様のことがいえる。（図5-4参照）

## 【0020】

データ数が多ければ、この最初の設定の4クロック (c l k) は、ほぼ無視できる値になり、アクセス時間は  $1/5$  に短縮されたことになる。

また、CPUのアドレッシングによる命令数は1命令×2データ=2命令と  $1/2$  にすることが可能である。データの R e a d, W r i t e 毎に転送は必要であるためFFT演算処理全体に占める転送時間の割合は大きい。

## 【0021】

262144点の入力データをFFT演算する場合について説明する。

この場合、262144点の実部・虚部のデータと131072点の回転子の実部・虚部の係数が、複数のローアドレスエリアにまたがって配置されているとする。

まず、FFT演算の総データをブロックに分ける。

同一ローアドレスエリアに512点の実部、虚部、回転子を格納できるとすると1ブロックのデータ数は512となり、512個のブロックができる。

512点 (A) のFFT演算を512回繰り返すことになる。

## 【0022】

周波数間引きで処理をする場合、まずビットリバースによる並び替えをする (FFT演算の一般的な並び替えである)。

並び替え後、実部 [0] ~ [262143]、虚部 [0] ~ [262143] を [0] 番目から512個、512点の実部、虚部データをワークエリアへ転送する。

この場合、FFT演算処理するため、ブロック数は2のm乗であることが必要である。

回転子のデータもワークエリアへ転送することで更に高速化が可能になる。

## 【0023】

第1ステージで、9段のFFT演算 (バタフライ演算: 図1の式1参照、2の9乗=512) を行い、再び、元の記憶エリアへ戻す。

次に実部、虚部のデータ [512] から512点のデータをワークエリアに転送し、再びFFT演算を実行する。

同様の処理を 5 1 2 個のブロックについて F F T 演算を繰り返す。(図 6 - 1 の 1st ステージ処理)

【 0 0 2 4 】

1st ステージ終了後、2nd ステージで使用する演算データの抽出のため実部、虚部データ [ 0 ] から 5 1 2 点毎にデータをワークエリアへ 5 1 2 個のデータを転送し、F F T 演算を行う。F F T 演算後、また元の記憶エリアへ転送する。

同様の処理を 5 1 2 個のブロックについて F F T 演算を繰り返す。(図 6 - 1 の 2nd ステージ処理)

【 0 0 2 5 】

このように、F F T 演算処理に伴うデータの転送において、ローアドレスが異なるためアクセス時間が掛かるが、本発明では、ローアドレスの設定は 1 回だけである。

従来の場合、残りの F F T 演算処理 9 段分、すなわち 9 倍の演算時毎にコマンドの再設定、ローアドレスの設定が必要であった。

【 0 0 2 6 】

上記 1st ステージ及び 2nd ステージの処理をフローチャートを図 6 - 1 示す。

また、分割された各ブロックの演算順序を図 6 - 2 に示す。

この処理により、F F T 演算において、データの実部、虚部、回転子の実部、虚部のデータの、演算器 ( レジスタ ) とメモリ間のデータ転送時間を従来にくらべて、大幅に短縮させることにより F F T 演算全体の高速化を図っている。

上記の如く、本発明のメモリに対するデータアクセスは、図 4 に示すように、演算デバイスのメモリには、同一ローアドレスでアクセス可能な範囲のデータをブロックとして記憶しておくことに特徴である。

【 0 0 2 7 】

上記の如き、2 6 2 1 4 4 点のデータによる F F T 演算では、全てのデータを図示するのが困難であるので、図 1 では、1 6 点のデータによる F F T 演算において、ブロックへの分割、並び替え、複数ステージ処理の採用による、本発明の F F T 演算高速アルゴリズムの概念図を図 1 に示す。

【 0 0 2 8 】

図1における、並び替え1では、第1ステージ後のFFT演算（図1では第2ステージのこと）では、使用データが前記第1ステージで使用したブロックとは別のブロックにまたがるので、データ選択（並び替え）によって使用データを選び直している。

図1では第3段、第4段に使用するデータをこの並び替え1で選び直している。

#### 【0029】

並び替え1以降は、再び、選び出した使用データ毎に演算ブロックを構成し、ブロック毎に転送し、FFT演算処理する。

上記の処理を、全てのブロックで繰り返すことによって、全てのFFT処理を完結させる。

この並び替えにより複数のステージを構成することで、同一のローアドレスエリア内にデータを配置して、FFT演算処理することが可能となる。

#### 【0030】

次に、回転子係数の選択であるが、周波数間引きによるFFT演算の場合、図9の表1の様な回転子を使用する。

回転子を入力データと掛け合わせて演算結果を得るため、回転子の係数もメモリからレジスタへ転送して演算される。

ということは、回転子の場合も、上記データ同様に、同一のローアドレスエリアに転送して演算することで転送時間の短縮が可能となる。

#### 【0031】

しかし、図9の表1のように、各段で使用する回転子は各ブロックで違うため、各ブロックの各ステージで使用する回転子係数を使用する順に並べて高速メモリへ転送して使用することで、回転子の違いを意識して使用する必要がなくなる。

各ブロックでは、図9の表2のように、テーブルの上から順に回転子を抽出して演算をすることにより、共通の演算ルーチンを組むことが可能になる。

この場合、回転子係数の抽出はテーブルの上から順に参照するという簡単な方法で実現可能であるため処理の高速化も容易に実現できる。

第2ステージでも同様に必要な回転子係数を抽出し、同一ローアドレスエリアへ転送することで高速化を図ることができる。

### 【0032】

また、現実的には、サンプリングしたデータのFFT周波数解析を行う場合、FFT演算の入力である実数部と虚数部データは、実数部にサンプリングデータを、虚数部に「0」を格納して演算する。虚数部が「0」であれば虚数データの演算を省略することができる。(図8の①：第1段の演算をを変更することにより実現でき、第1ステージの各ブロックで共通に使用することが可能)

### 【0033】

実際には、 $X_0 \sim X_{15}$  及び  $Y_0 \sim Y_{15}$  のデータと回転子は、

例えば、

$$X_1 = X_{r1} (\text{実部}) + j X_{i1} (\text{虚部})、$$

$$X_2 = X_{r2} (\text{実部}) + j X_{i2} (\text{虚部})、$$

$$Y_1 = Y_{r1} (\text{実部}) + j Y_{i1} (\text{虚部})、$$

$$Y_2 = Y_{r2} (\text{実部}) + j Y_{i2} (\text{虚部})、$$

$$W = W_r (\text{実部}) + j W_i (\text{虚部}) \text{ の複素数で表される。}$$

よって、周波数間引き法における前記演算式1は、

$$\begin{aligned} Y_1 &= X_1 + X_2 = (X_{r1} + jX_{i1}) + (X_{r2} + jX_{i2}) \\ &= (X_{r1} + X_{r2}) + j(X_{i1} + X_{i2}) \dots \dots \dots (1-1) \end{aligned}$$

$$\begin{aligned} Y_2 &= (X_1 - X_2) W = \{ (X_{r1} + jX_{i1}) - (X_{r2} + jX_{i2}) \} (W_r + jW_i) \\ &= \{ (X_{r1} - X_{r2}) + j(X_{i1} - X_{i2}) \} (W_r + jW_i) \end{aligned}$$

$X_R = X_{r1} - X_{r2}$ ,  $X_I = X_{i1} - X_{i2}$  とすると

$$Y_2 = (X_R + jX_I) (W_r + jW_i) = (X_R \cdot W_r - X_I \cdot W_i) + j(X_R \cdot W_i + X_I \cdot W_r) \dots (1-2)$$

(1-1)より、

$$Y_{r1} (\text{実部}) = X_{r1} + X_{r2}$$

$$Y_{i1} (\text{虚部}) = X_{i1} + X_{i2}$$

(1-2)より、

$$Y_{r2} (\text{実部}) = X_R \cdot W_r - X_I \cdot W_i$$

$$Yi2 \text{ (虚部)} = XR \cdot Wi + XI \cdot Wr$$

実部、虚部に対して上記の演算を実行する。

ここで、サンプリングデータとして実部のデータのみが与えられると、虚部が「0」で演算されるため、

$$Yr1 \text{ (実部)} = Xr1 + Xr2$$

$$Yi1 \text{ (虚部)} = Xi1 + Xi2 = 0 + 0$$

$$Yr2 \text{ (実部)} = XR \cdot Wr - XI \cdot Wi = XR \cdot Wr$$

$$Yi2 \text{ (虚部)} = XR \cdot Wi + XI \cdot Wr = XR \cdot Wi$$

になる。これは、一段目に限ってであり、二段目以降は虚部のデータが存在するので、正規の演算が必要になる。

#### 【 0 0 3 4 】

また、回転子の値が実部 = 1、虚部 = 0、または、実部 = 0、虚部 = 1 であることにより回転子と演算データの乗算を省略する事ができる（図 8 の②：第 3 段、第 4 段の演算を変更することにより実現でき、第 2 ステージで使用する回転子が同じなので処理ルーチンを共通に使用することが可能）とか、等の既知の高速化手法を反映させることが容易であり、これにより更に高速化が可能になる。

例えば、回転子  $W = (Wr, Wi) = (1, 0)$  の場合、

$$Yr2 \text{ (実部)} = XR \cdot Wr - XI \cdot Wi = XR$$

$$Yi2 \text{ (虚部)} = XR \cdot Wi + XI \cdot Wr = XI$$

また、回転子  $W = (Wr, Wi) = (0, j)$  の場合、

$$Yr2 \text{ (実部)} = XR \cdot Wr - XI \cdot Wi = -XI$$

$$Yi2 \text{ (虚部)} = XR \cdot Wi + XI \cdot Wr = XR$$

として、演算を省略することが可能である。

#### 【 0 0 3 5 】

ペンティアム CPU を使った一般的なパーソナルコンピュータを使用して 2 6 2 1 4 4 点の FFT を行った場合、本発明を適用しない場合に比べ、本発明を適用することで FFT の演算時間を数分の 1 に縮める事が可能である。

また、本発明における処理は、厳密にローアドレスエリア内に設定する必要もない。

2 6 2 1 4 4 点 F F T 演算に対して、十分小さい 5 1 2 点とか、1 0 2 4 点のブロックに分割することで、ロードアドレスの再設定とか、バンクの再設定の頻度が減るため高速に処理可能となる。

また、第 1 ステージ、第 2 ステージの F F T 演算データ数は同じにする必要はなく、2 の m 乗、2 の k 乗と、F F T 演算可能な数にまとまっていればよい。ステージも、2 分割するのではなく、ステージ数を 3 又は 4 とする事もできる。

【 0 0 3 6 】

【発明の効果】

請求項 1 に記載の発明では、多数の被 F F T データを F F T 演算する高速 F F T 演算処理方法において、前記多数の被 F F T データ (N) を、F F T 演算処理に用いるメモリのアクセスに適したブロックに分割するステップ(a)と、

前記メモリに対して、前記ブロックに分割されたデータを順次転送するステップ(b)と、前記メモリに転送された F F T データによる F F T 演算処理を実行するステップ(c)と、前記ステップ(c)及びステップ(c)を、全ての分割されたブロックについて繰り返すステップ(d)とで、高速 F F T 演算処理を実行することにより、

F F T 演算にともなって生じる、メモリから演算レジスタへの演算毎のデータの転送時間を短縮することにより、F F T 演算全体の処理時間を短縮することができる。

【 0 0 3 7 】

また、請求項 2 に記載の発明では、前記多数の被 F F T データの F F T 演算処理を複数のステージに分割し、各ステージ間で、被 F F T データの並び替えを実行するステップを有することにより、より少ないメモリで F F T 演算が可能になる。

また、請求項 3 に記載の発明では、前記ステップ(a)における被 F F T データの分割は、前記メモリアクセスに際して、メモリを構成するバンクの再設定が不要なデータ範囲内になるように行うことにより、バンク構成のメモリの場合に、バンクの再設定なしにデータのアクセスが可能になるので、メモリから演算レジスタへの演算毎のデータの転送時間を短縮することにより、F F T 演算全体の処

理時間を短縮することができる。

また、請求項 4 に記載の発明では、前記ステップ(a)における被 F F T データの分割は、前記メモリアクセスに際して、ローアドレス又はカラムアドレスの再設定が不要なデータ範囲内になるように行うことにより、一度設定した、ローアドレス又はカラムアドレスの再設定なしに、データのアクセスが可能になるので、メモリから演算レジスタへの演算毎のデータの転送時間を短縮することにより、F F T 演算全体の処理時間を短縮することができる。

【 0 0 3 8 】

また、請求項 5 に記載の発明では、前記被 F F T データは、それぞれ実数部及び虚数部からなり、回転子との間で F F T 演算処理される。

また、請求項 6 に記載の発明では、前記回転子が前記ブロックに対応して予めテーブルに保管されていることにより、より高速な F F T 演算処理が可能になる。

【 0 0 3 9 】

また、請求項 7 に記載の発明では、前記被 F F T データの内の虚数部データの演算を省略する。

また、請求項 8 に記載の発明では、前記 F F T 演算において、回転子の実数部又は虚数部のいずれか一方がゼロの場合には、乗算を省略することにより、更に高速の F F T 演算処理が可能になる。

この場合、現実的な信号解析において、サンプリングしたデータの周波数解析を行う場合、虚数データの演算を省略するとか、回転子の値が実部 = 1，虚部 = 0、または、実部 = 0、虚部 = 1 であることにより乗算を省略する事ができるとか、等の既知の高速化手法を反映させやすい事を特徴とする。

【図面の簡単な説明】

【図 1】

本発明の概念を示すブロック図である。

【図 2】

従来 F F T 演算処理を示す図である。



【図 3】

従来のデータアクセスの概略図である。

【図 4】

本発明によるデータアクセスの概略図である。

【図 5】

メモリアクセス方式のリードタイミングを示す図である

【図 6】

本発明のフローチャート及びブロックの処理順を示す図である。

【図 7】

本発明の、FFT回転子の選択を示した図である

【図 8】

既知の高速手法を取り入れた高速処理概略図

【図 9】

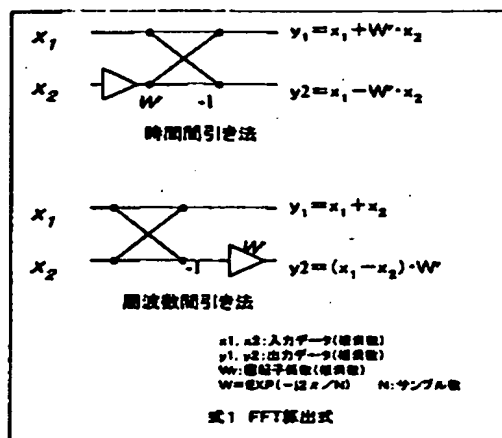
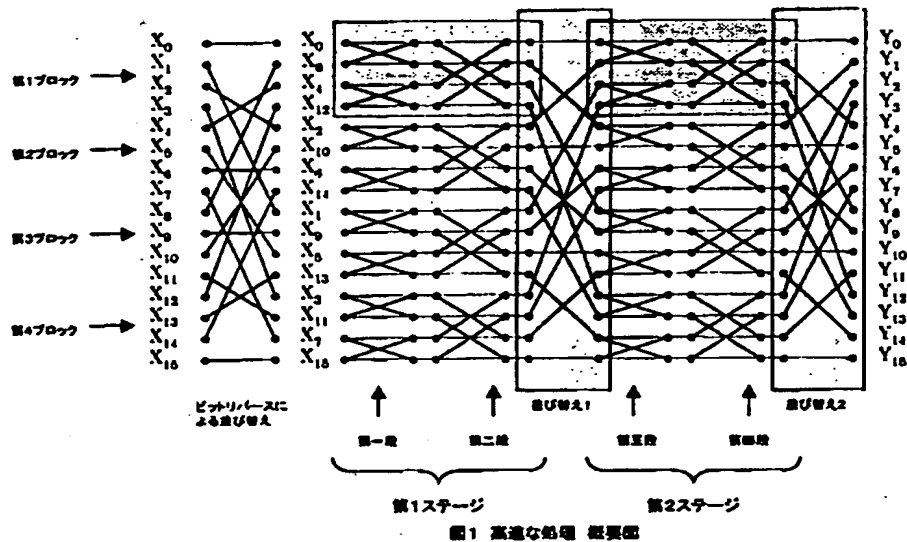
本発明の、FFT回転子の選択及びテーブルを示した表である

【符号の説明】

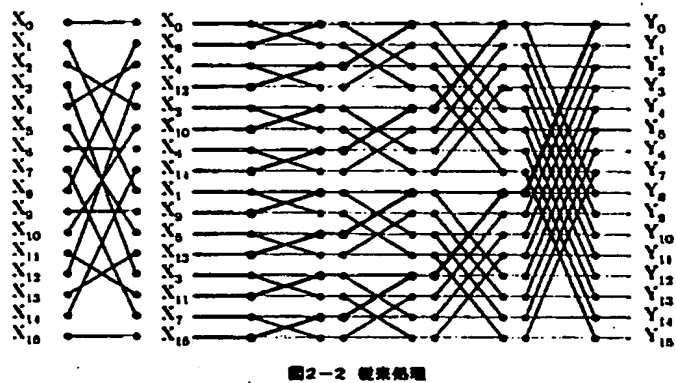
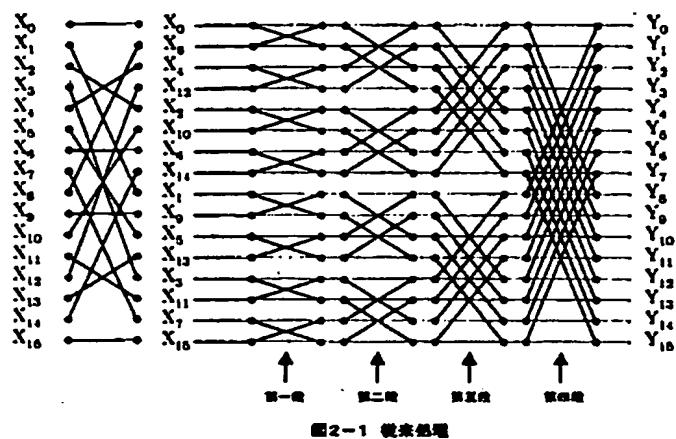
なし

【書類名】 図面

【図 1】



【図 2】



【図 3】

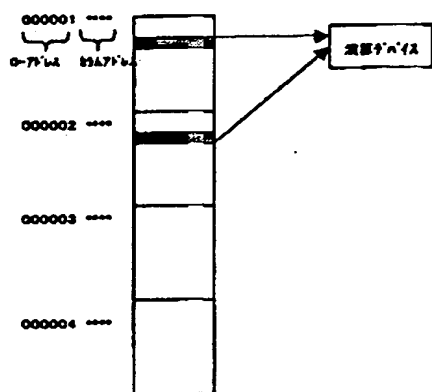


図3 従来方式によるデータアクセス概略図

【図 4】

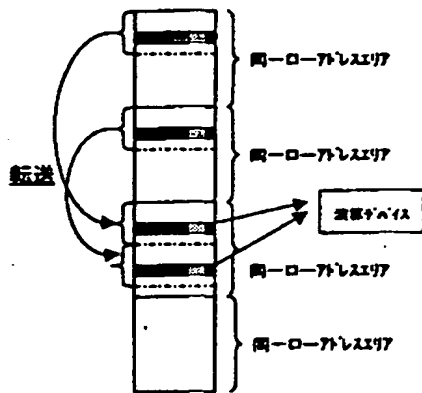
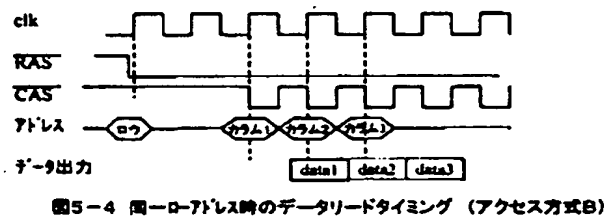
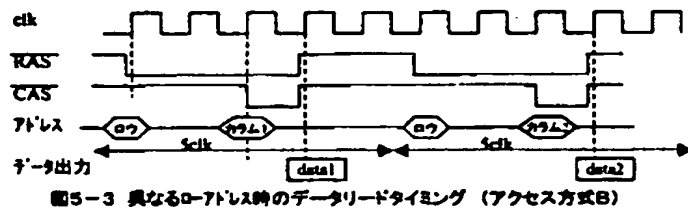
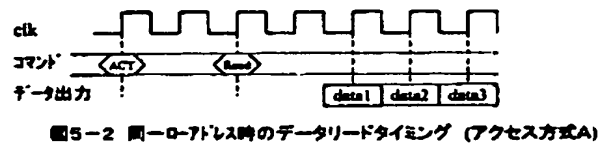
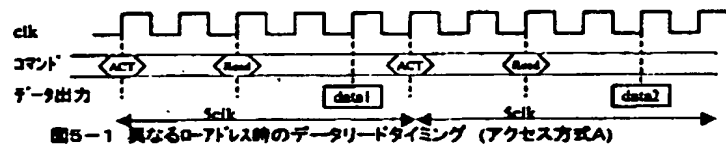


図 4 本発明によるデータアクセスの概略図

【図 5】



【図 6】

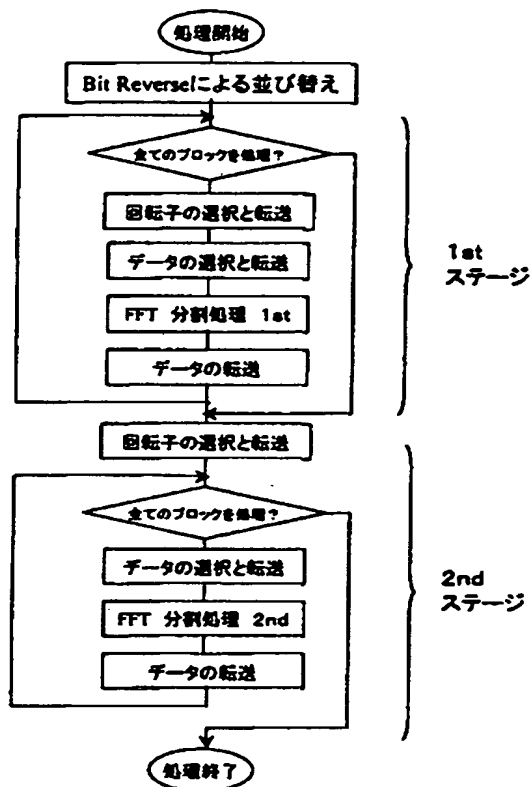


図6-1 フローチャート

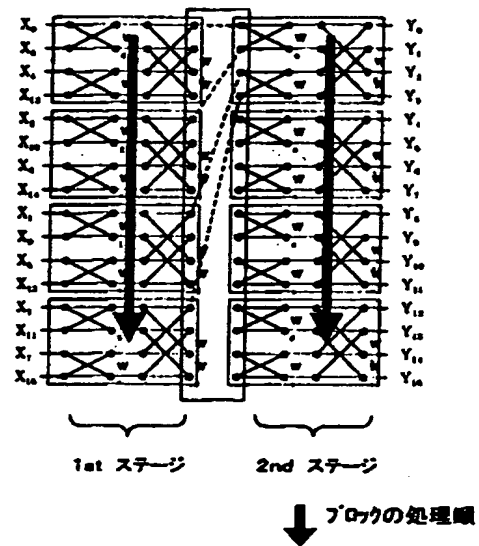
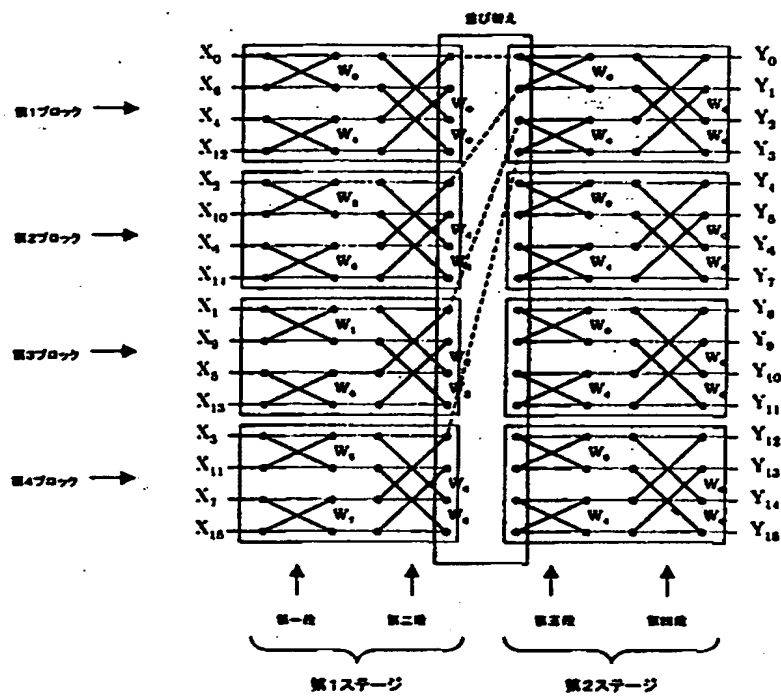
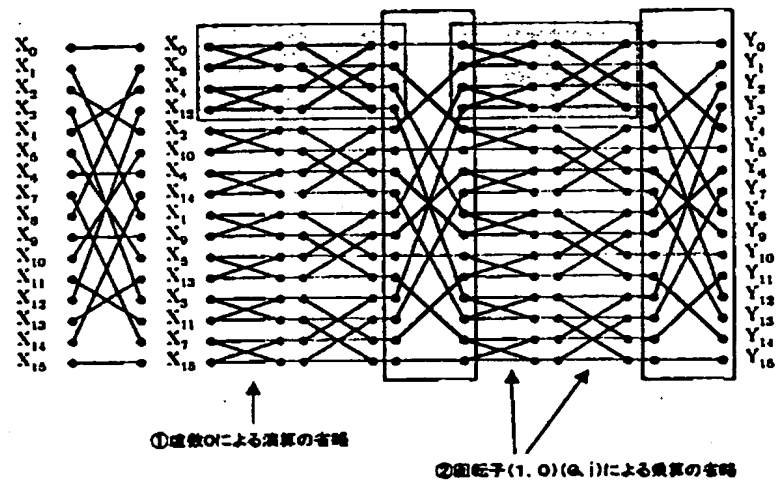


図6-2 ブロックの処理順

【図 7】



【図 8】



【図 9】

第1ブロック				第2ブロック			
第一番	第二番	第三番	第四番	第一番	第二番	第三番	第四番
w[0]	w[0]	w[0]	w[0]	w[0]	w[0]	w[0]	w[0]
w[1]	w[1]	w[1]	w[1]	w[1]	w[1]	w[1]	w[1]
w[2]	w[2]	w[2]	w[2]	w[2]	w[2]	w[2]	w[2]
w[3]	w[3]	w[3]	w[3]	w[3]	w[3]	w[3]	w[3]
w[4]	w[4]	w[4]	w[4]	w[4]	w[4]	w[4]	w[4]
w[5]	w[5]	w[5]	w[5]	w[5]	w[5]	w[5]	w[5]
w[6]	w[6]	w[6]	w[6]	w[6]	w[6]	w[6]	w[6]
w[7]	w[7]	w[7]	w[7]	w[7]	w[7]	w[7]	w[7]
第3ブロック				第4ブロック			
第一番	第二番	第三番	第四番	第一番	第二番	第三番	第四番
w[0]	w[0]	w[0]	w[0]	w[0]	w[0]	w[0]	w[0]
w[1]	w[1]	w[1]	w[1]	w[1]	w[1]	w[1]	w[1]
w[2]	w[2]	w[2]	w[2]	w[2]	w[2]	w[2]	w[2]
w[3]	w[3]	w[3]	w[3]	w[3]	w[3]	w[3]	w[3]
w[4]	w[4]	w[4]	w[4]	w[4]	w[4]	w[4]	w[4]
w[5]	w[5]	w[5]	w[5]	w[5]	w[5]	w[5]	w[5]
w[6]	w[6]	w[6]	w[6]	w[6]	w[6]	w[6]	w[6]
w[7]	w[7]	w[7]	w[7]	w[7]	w[7]	w[7]	w[7]

表1 回転子の選択

	第1ブロック	第2ブロック	第3ブロック	第4ブロック
0	w[0]	w[2]	w[1]	w[3]
1	w[4]	w[6]	w[5]	w[7]
2	w[0]	w[4]	w[2]	w[6]

表2 第1ステージで使用する回転子のテーブル

【書類名】 要約書

【要約】

【課題】 F F T 演算にともなって生じる、メモリから演算レジスタへの演算毎のデータの転送時間を短縮することにより、F F T 演算全体の処理時間を短縮することである。

【解決手段】 多数の被 F F T データを F F T 演算する高速 F F T 演算処理方法において、前記多数の被 F F T データ (N) を、F F T 演算処理手段内のメモリのアクセスに適したブロックに分割するステップ(a)と、前記メモリに対して、前記ブロックに分割されたデータを順次転送するステップ(b)と、前記メモリに転送された F F T データによる F F T 演算処理を実行するステップ(c)と、前記ステップ(c)及びステップ(c)を、全ての分割されたブロックについて繰り返すステップ(d)と、を有することを特徴とする高速 F F T 演算処理方法。

【選択図】 図 1



特 2001-002029

認定・付加情報

特許出願の番号	特願 2001-002029
受付番号	50100013674
書類名	特許願
担当官	第三担当上席 0092
作成日	平成13年 1月16日

<認定情報・付加情報>

【提出日】	平成13年 1月10日
-------	-------------

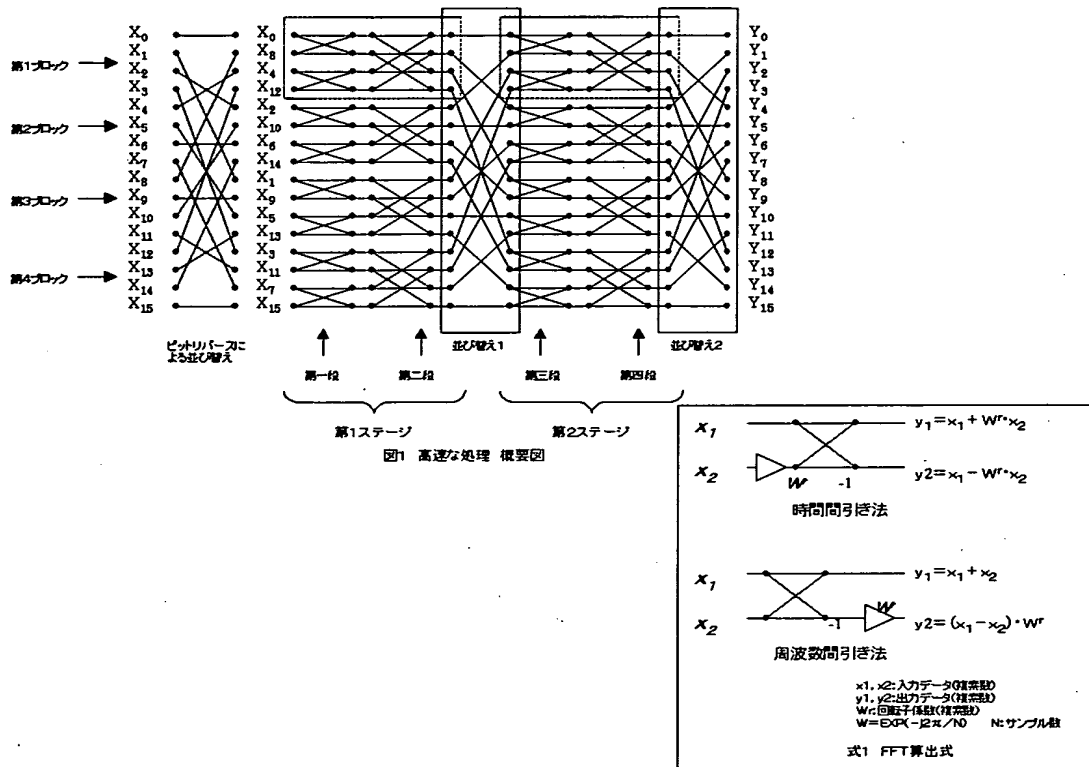
次頁無

【書類名】 手続補正書  
【整理番号】 S00-12-17  
【あて先】 特許庁長官 殿  
【事件の表示】  
    【出願番号】 特願2001- 2029  
【補正をする者】  
    【識別番号】 000117744  
    【氏名又は名称】 安藤電気株式会社  
【代理人】  
    【識別番号】 100099195  
    【弁理士】  
    【氏名又は名称】 宮越 典明  
【手続補正 1】  
    【補正対象書類名】 図面  
    【補正対象項目名】 全図  
    【補正方法】 変更  
    【補正の内容】 1  
【ブルーフの要否】 要

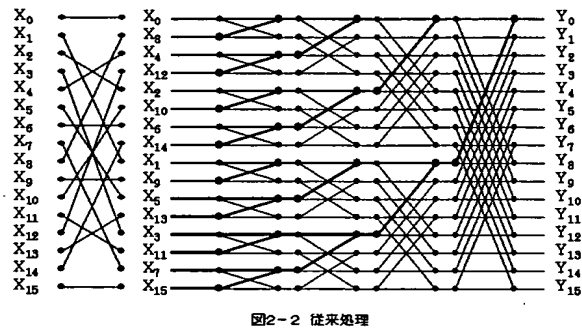
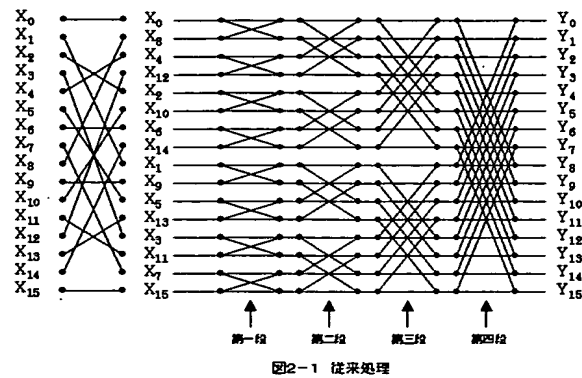
【書類名】

図面

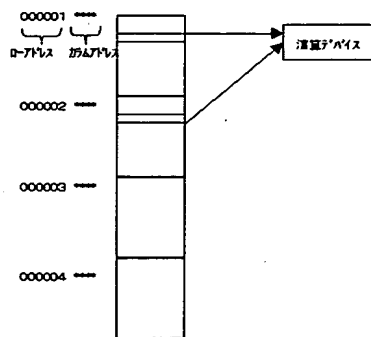
【図 1】



【図 2】



【図 3】



【図 4】

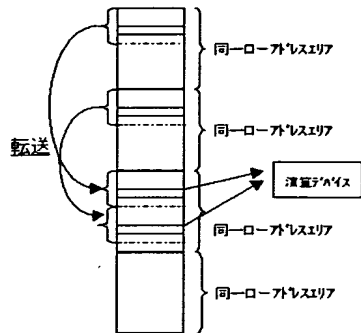


図4 本発明によるデータアクセスの概略図

【図 5】

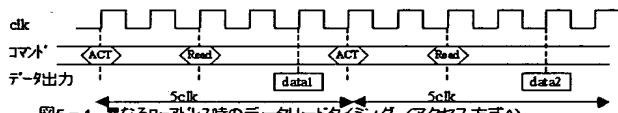


図5-1 異なるローアドレス時のデータリードタイミング (アクセス方式A)

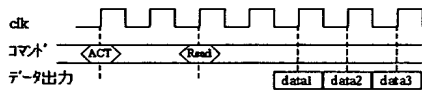


図5-2 同一ローアドレス時のデータリードタイミング (アクセス方式A)

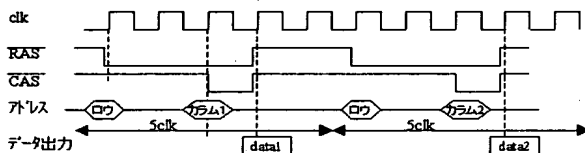


図5-3 異なるローアドレス時のデータリードタイミング (アクセス方式B)

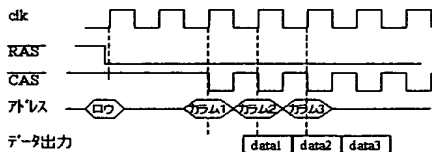
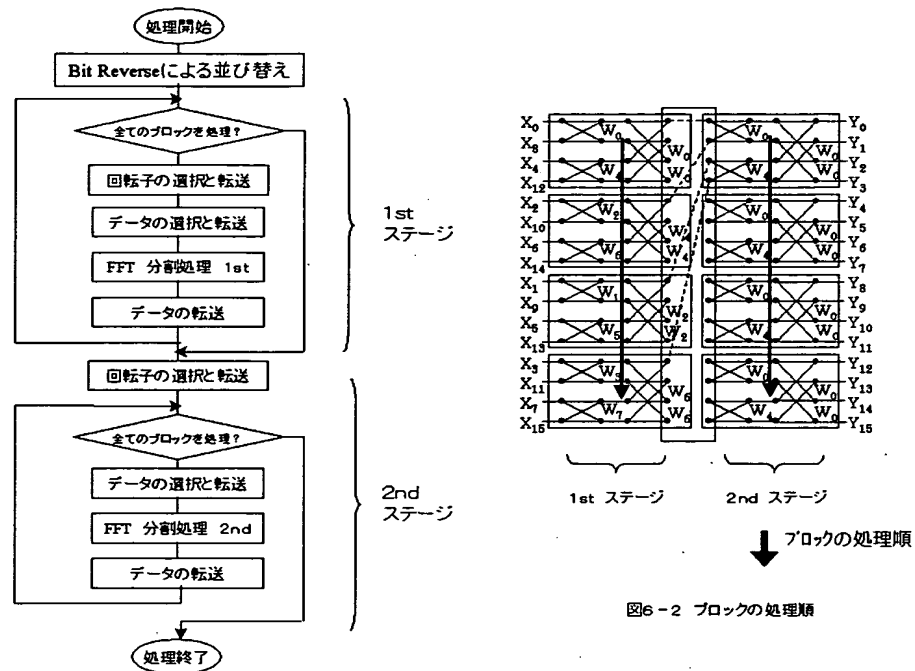


図5-4 同一ローアドレス時のデータリードタイミング (アクセス方式B)

【図 6】



【図 7】

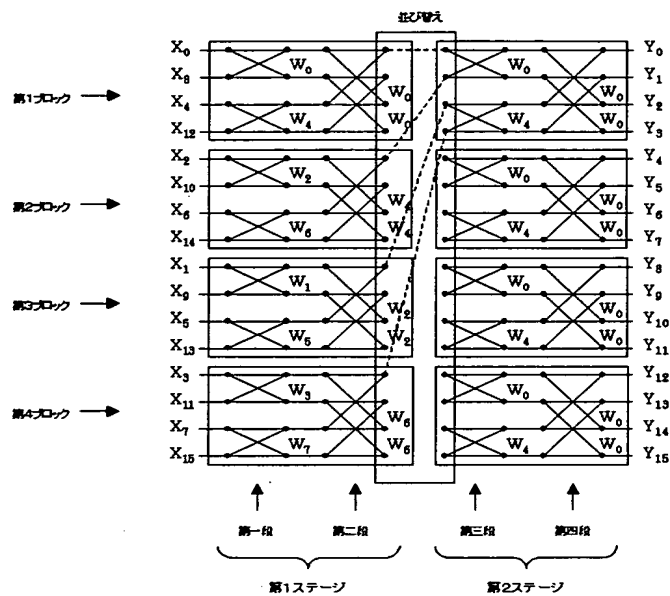


図7 回転子の選択

【図 8】

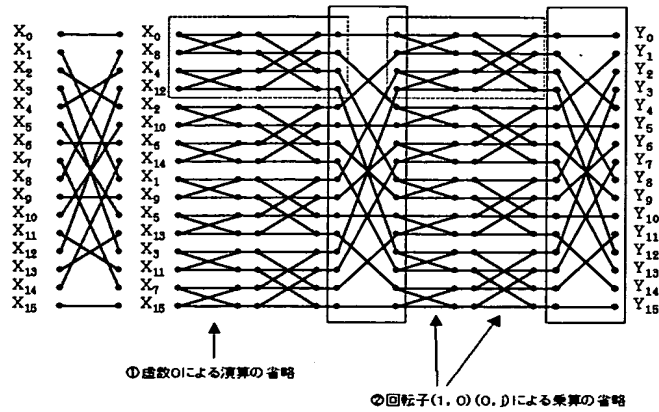


図8 既知の高速手法をとつた高速処理 概略図



【図 9】

第1ブロック				第2ブロック			
第一段	第二段	第三段	第四段	第一段	第二段	第三段	第四段
W[0]	W[0]	W[0]	W[0]	W[0]	W[0]	W[0]	W[0]
W[4]	W[4]	W[4]	W[4]	W[4]	W[4]	W[4]	W[4]
W[2]	W[2]	W[2]	W[2]	W[2]	W[2]	W[2]	W[2]
W[6]	W[6]	W[6]	W[6]	W[6]	W[6]	W[6]	W[6]
W[1]	W[1]	W[1]	W[1]	W[1]	W[1]	W[1]	W[1]
W[5]	W[5]	W[5]	W[5]	W[5]	W[5]	W[5]	W[5]
W[3]	W[3]	W[3]	W[3]	W[3]	W[3]	W[3]	W[3]
W[7]	W[7]	W[7]	W[7]	W[7]	W[7]	W[7]	W[7]
第3ブロック				第4ブロック			
第一段	第二段	第三段	第四段	第一段	第二段	第三段	第四段
W[0]	W[0]	W[0]	W[0]	W[0]	W[0]	W[0]	W[0]
W[4]	W[4]	W[4]	W[4]	W[4]	W[4]	W[4]	W[4]
W[2]	W[2]	W[2]	W[2]	W[2]	W[2]	W[2]	W[2]
W[6]	W[6]	W[6]	W[6]	W[6]	W[6]	W[6]	W[6]
W[1]	W[1]	W[1]	W[1]	W[1]	W[1]	W[1]	W[1]
W[5]	W[5]	W[5]	W[5]	W[5]	W[5]	W[5]	W[5]
W[3]	W[3]	W[3]	W[3]	W[3]	W[3]	W[3]	W[3]
W[7]	W[7]	W[7]	W[7]	W[7]	W[7]	W[7]	W[7]

表1 回転子の選択

	第1ブロック	第2ブロック	第3ブロック	第4ブロック
0	W[0]	W[2]	W[1]	W[3]
1	W[4]	W[6]	W[5]	W[7]
2	W[0]	W[4]	W[2]	W[6]

表2 第1ステージで使用する回転子のテーブル

特2001-002029

## 認定・付加情報

特許出願の番号	特願2001-002029
受付番号	50100026632
書類名	手続補正書
担当官	喜多川 哲次 1804
作成日	平成13年 1月17日

### <認定情報・付加情報>

【提出日】	平成13年 1月11日
-------	-------------

次頁無

出 願 人 履 歴 情 報

識別番号 [000117744]

1. 変更年月日 1990年 8月10日  
[変更理由] 新規登録  
住 所 東京都大田区蒲田4丁目19番7号  
氏 名 安藤電気株式会社
2. 変更年月日 2001年 4月13日  
[変更理由] 住所変更  
住 所 東京都大田区蒲田五丁目29番3号  
氏 名 安藤電気株式会社